PYTHON笔记

当有多个参数需要接收，但是不知道参数有多少个的时候，有两种方式可以：

1、将不知道有多少的参数（可变参数）放入list处理，定义可变参数n接收这些参数，这些参入将会被封装为一个tuple保存在n中，一般写法是\*args （可变参数）

可以定义可变参数接受多个变量，比如

def sum\_n(\*n):

sum = 0;

for i in n:

print(i)

sum = sum + int(i)\*int(i);

print(sum);

sum\_n(1, 3, 5, 7);

python会自动将1,3,5,7组装为一个tuple(1, 3, 5, 7)

如果函数用可变参数接收数据，且调用函数时想要传入list或tuple时：

def sum\_n(\*n):

sum = 0;

for i in n:

print(i)

sum = sum + int(i)\*int(i);

print(sum);

sum\_n(\*[1, 3, 5, 7]);

可以在实参前面加\*，表示传进去的是list里的参数

2、将不知道有多少的参数（可变参数）放入dict处理，定义可变参数kw接收这些参数，这些参数会被封装进dict，等号前面为key，等号后面为value，一般写法是\*\*kw（关键字参数）

def person(name,age,\*\*kw):

print('name :', name);

print('age :', age);

print('other paramters :', kw);

person('Bob', 5, city='chengdu', height='175')

输出：

name : Bob

age : 5

other paramters : {'city': 'chengdu', 'height': '175'}

3、当输入的参数必须指定key的时候，使用命名关键字参数；\*作为分割符，\*后面的是命名关键字参数 （命名关键字参数）

def person(name,age,\*,city,job):

print('name :', name);

print('age :', age);

print('city :', city);

print('job :', job);

person('Bob', 5, city='chengdu', job='worker')

当参数中间有可变参数时，后面的默认为命名关键字参数

def person(name,age,\*arg,city,job):

print('name :', name);

print('age :', age);

print('arg :', arg)

print('city :', city);

print('job :', job);

person('Bob', 5, 'aaa','bbb','ccc' ,city='chengdu', job='worker')

输出：

name : Bob

age : 5

arg : ('aaa', 'bbb', 'ccc')

city : chengdu

job : worker

**类与对象：**

type()可以查看对象类型

定义对象：

class Car:

brand = ‘Benz’ #类的属性

country = ‘Germany’

def pressHorn(): #类的方法

print(‘didi’)

实例属性通常在类的初始化方法\_\_init\_\_中定义，类在初始化时，解释器会去执行

def \_\_init\_\_(**self**):

self.color = “red”

self.engine = “612485”

**当调用初始化方法实例化对象时，解释器首先会在内存中创建一个类的实例对象，解释器就把这个对象本身作为参数传递进变量self中**

**异常捕获：**

try:

….

except XXXError:

….

当出现多种except类型时，

try:

….

except ZeroDivisionError:

….

except IndexError:

….

捕获所有类型的异常：

import traceback:

try:

100/0

except Exception as e:

print(“未知异常”,e)

print(traceback.format\_exc())